Slow Memory Access

PowerPC system tables

Tue, Feb 19, 2002

Slow access to nonvolatile memory on the PowerPC version of the system may require some reevaluation of how the system code should operate. This note discusses some possible modifications to the system inspired by recognition of such slow access. The access time to nonvolatile memory that resides on a PMC board in the 2401 CPU board is about 1 μs. For a 233 MHz cpu clock speed, this access time is a big deal.

Most system tables reside in nonvolatile memory, because their contents need to be preserved across system resets, including power outages. Consideration should be given to those aspects of system operation that require frequent access to such tables. The ADATA table is an example, as it is used heavily during alarm scanning. When the cost of scanning the entire ADATA table, looking for entries that need alarm checking, was realized, the alarm scan logic was rewritten to minimize accesses to this table. The first step was to maintain a table allocated from volatile memory to keep a list of all channels that are currently in the alarm scan. The new alarm scan logic processed each such channel number in turn; in this way, only ADATA entries were examined that needed to be checked for alarm conditions. An additional change accessed the ADATA fields as 32-bit long integers, only doing so when necessary. For example, the alarm flags and trip count words can be accessed by a single 32-bit access, with the copies used during the alarm checking logic. Only if either word was altered was it necessary to write back the 32-bit value to the nonvolatile fields of the ADATA entry. The nominal and tolerance fields could also be accessed with a single 32-bit read. By following such methods, the number of accesses to nonvolatile memory was minimized. The same optimization techniques were also applied for the digital bit scan and comment scan, with the result that the complete alarm scan is now executed in less than 1 ms.

The CINFO table is used to house additional information, not covered by fields in the ADESC table, that is only needed by a small number of analog channels. So far, it has only been used to hold fast digitizer parameters. Each entry occupies some multiple of 8 bytes, with only 8 and 16 byte entries needed to date. Each entry begins with three fields, as follows:

Field	*Size*	*Meaning*
`size`	1	Size of this entry, where 00 signifies 8 by default
`type`	1	Entry type = 1, 2, 3, etc
`chan`	2	Analog channel number

The primary user of this table is LOOPFTPM, the local application that supports the Acnet FTPMAN protocol that is used for accessing waveform data for plotting purposes. The access to the table is provided by a routine such as `CINFOentry`, whose arguments are a type number and a channel number. This function returns the pointer to the CINFO entry that matches it in `type` and in `chan`. (If no matching entry is found, it returns NULL.) This logic is convenient for LOOPFTPM, but it is inherently inefficient when access to slow memory is considered. When this was discovered, the declared size of CINFO was dramatically reduced, as it was set to 512 eight-byte entries by default, so that every time a (failed) search was performed, 512 accesses had to be made. Worse than that, each entry needed to be examined for the `type`, `chan`, and `size`, resulting in three accesses per entry. Of the 25 PowerPC nodes involved, only three used any CINFO entries at all, and their needs were for a small fraction of the 512 entries allowed. This was a quick fix for the problem, but a rewrite of the LOOPFTPM logic is probably needed, in order to improve its efficiency in light of slow access to nonvolatile memory.

The LISTP table is set up completely during system initialization, so that it has no need for being nonvolatile. It is used to obtain a "list number," which is a kind of message-id that can be associated with an active data request in order to assist in associating a reply message with the request support structure. It is mostly accessed by table look-up rather than searches, so that its cost of being accessed as slow memory is not significant.

The CODES table is a kind of directory of all the current downloaded program files. It is usually accessed by searching. During Data Access Table processing at 15 Hz, each active local application instance found in the LATBL system table requires a search of CODES to get the pointer to the executable code for that local application. There may be some 64 entries in a typical system's CODES table. The search code checks the 4-character PROG name first, before it checks the TYPE name (such as 'LOOP') so that means that visiting each entry would usually require only a single memory access. Maybe it's not so bad.

The IPNAT is searched for a match on an IP address, in order to come up with a node number, or it is search for a node number in order to retrieve the corresponding IP address. In a sense, it is a kind of DNS cache. The local application LOOPDNSQ manages the DNS queries that keep the table contents from growing stale. Each entry independently times out in about 8 hours, and the DNS server is queried to get an up date on its official IP address. In order for this to work, all nodes are registered with the DNS with the name 'nodexxxx', where xxxx is the node number in hexadecimal. All such registered names now in use are of the form node05xx or node06xx. At Fermilab, an example is node051A.fnal.gov.